On synchronous robotic networks
Part I: Models, tasks and complexity notions

Sonia Martínez Francesco Bullo Jorge Cortés Emilio Frazzoli

***Abstract*— This paper proposes a formal model for a network of robotic agents that move and communicate. Building on concepts from distributed computation, robotics and control theory, we define notions of robotic network, control and communication law, coordination task, and time and communication complexity. We illustrate our model and compute the proposed complexity measures in the example of a network of locally connected agents on a circle that agree upon a direction of motion and pursue their immediate neighbors.**

I. INTRODUCTION

Problem motivation: The study of networked mobile systems presents new challenges that lie at the confluence of communication, computing, and control. In this paper we consider the problem of designing joint communication protocols and control algorithms for groups of agents with controlled mobility. For such groups of agents we define the notion of communication and control law by extending the classic notion of distributed algorithm in synchronous networks. Decentralized control strategies are appealing for networks of robots because they can be scalable and they provide robustness to vehicle and communication failures.

One of our key objectives is to develop a computable theory of time and communication complexity for motion coordination algorithms. Hopefully, our formal model will be suitable to analyze objectively the performance of various coordination algorithms. It is our contention that such a theory is required to assess the complex trade-offs between computation, communication and motion control or, in other words, to establish what algorithms are *scalable* and practically implementable in large networks of mobile autonomous agents. The need for modern models of computation in wireless and sensor network applications is discussed in the well-known report [3].

Literature review: To study complexity of motion coordination, our starting points are the standard notions of *synchronous and asynchronous networks* in distributed and parallel computation, e.g., see Lynch [4] and, with an emphasis on numerical methods, Bertsekas and Tsitsiklis [5]. This established body of knowledge, however, is not applicable to the robotic network setting because of the agents' mobility and the ensuing dynamic communication topology.

An important contribution towards a network model of mobile interacting robots is introduced by Suzuki and Yamashita [6], see also [7], [8]. The Suzuki-Yamashita model consists of a group of "distributed anonymous mobile robots" that interact by sensing each other's relative position. A related model is presented in [9], [10]. A brief survey of models, algorithms, and the need for appropriate complexity notions is presented in [11].

Recently, a notion of communication complexity for control and communication algorithms in multi-robot systems is analyzed in [12], see also [13] where a formal model of communication and control laws for multi-agent networks is proposed. A general modeling paradigm is discussed in [14]. The time complexity of a distributed algorithm for coordinated motion planning is computed in [15].

Statement of contributions: We summarize our approach as follows. A *robotic network* is a group of robotic agents moving in space and endowed with communication capabilities. The agents position obey a differential equation and the communication topology is a function of the agents' relative positions. Each agent repeatedly performs communication, computation and physical motion as described next. At predetermined time instants, the agents exchange information along the communication graph and update their internal state. Between successive communication instants, the agents move according to a motion control law, computed as a function of the agent location and of the available information gathered through communication with other agents. In short, a *control and communication law* for a robotic network consists of a message-generation function (what do the agents communicate?), a state-transition function (how do the agents update their internal state with the received information?), and a motion control law (how do the agents move between communication rounds?). We then define the notion of *time complexity* of a control and communication law (aimed at solving a given coordination task) as the minimum number of communication rounds required by the agents to achieve the task. The *time complexity of a coordination task* is the minimum time complexity of any algorithm achieving the task. We also provide similar definitions for mean and total communication complexity. We show that our notions of complexity satisfy a basic well-posedness property that we refer to as "invariance under reschedulings." We illustrate these concepts and results in a network of locally connected agents evolving on the circle. We define a control and

The complete version of this work is [1]. This paper is submitted to the 2005 CDC jointly with [2].

Sonia Martínez and Francesco Bullo are with the Department of Mechanical and Environmental Engineering, University of California at Santa Barbara, Santa Barbara, California 93106, {smartine,bullo}@engineering.ucsb.edu

Jorge Cortés is with the Department of Applied Mathematics and Statistics, University of California at Santa Cruz, Santa Cruz, California 95064, jcortes@ucsc.edu

Emilio Frazzoli is with the Department of Mechanical and Aerospace Engineering, University of California at Los Angeles, Los Angeles, California 90095, frazzoli@ucla.edu

communication law for this network that achieves consensus on the agents' direction of motion and equidistance between the agents' positions. Furthermore, we provide upper and lower bounds on the time and communication complexity to achieve these tasks with the proposed law. The companion paper [2] builds on this framework to establish complexity estimates for a variety of motion coordination algorithms that achieve rendezvous and deployment.

Notation: We let $\texttt{BooleSet}$ be the set $\{\texttt{true}, \texttt{false}\}$. We let $\prod_{i\in\{1,\dots,N\}} S_i$ denote the Cartesian product of sets $S_1,\dots,S_N$. We let $\mathbb{R}_+$ and $\overline{\mathbb{R}}_+$ denote the set of strictly positive and non-negative real numbers, respectively. The set of positive natural numbers is denoted by $\mathbb{N}$ and $\mathbb{N}_0$ denotes the set of non-negative integers. If S is a set, then $\operatorname{diag}(S\times S) = \{(s,s)\in S\times S \mid s\in S\}$. For $x\in\mathbb{R}$, we let $\lfloor x\rfloor$ denote the floor of x. For $x\in\mathbb{R}^d$, we denote by $\|x\|_2$ and $\|x\|_\infty$ the Euclidean and the ∞-norm of x, respectively. Recall that $\|x\|_\infty \le \|x\|_2 \le \sqrt{d}\|x\|_\infty$ for all $x\in\mathbb{R}^d$. For $f,g\colon\mathbb{N}\to\mathbb{R}$, we say that $f\in O(g)$ (respectively, $f\in\Omega(g)$) if there exist $N_0\in\mathbb{N}$ and $k\in\mathbb{R}_+$ such that $|f(N)|\le k|g(N)|$ for all $N\ge N_0$ (respectively, $|f(N)|\ge k|g(N)|$ for all $N\ge N_0$). If $f\in O(g)$ and $f\in\Omega(g)$, then we use the notation $f\in\Theta(g)$.

II. A formal model for synchronous robotic networks

Here we introduce a notion of robotic network as a group of robotic agents with the ability to move and communicate according to a specified communication topology.

A. The physical components of a robotic network

Here we introduce our basic definition of physical quantities such as the agents and such as the ability of agents to communicate. We begin by providing a basic model for how each robotic agent moves in space. A *control system* is a tuple (X,U,X_0,f) consisting of

(i) X is a differentiable manifold, called the *state space*;
(ii) U is a compact subset of $\mathbb{R}^m$ containing 0, called the *input space*;
(iii) X_0 is a subset of X, called the *set of allowable initial states*;
(iv) $f\colon X\times U\to TX$ is a C^∞-map with $f(x,u)\in T_xX$ for all $(x,u)\in X\times U$.

We refer to $x\in X$ and $u\in U$ as a *state* and an *input* of the control system, respectively. We will often consider control-affine systems, i.e., control systems with $f(x,u) = f_0(x)+\sum_{a=1}^m f_a(x)\,u_a$. In such a case, we represent f as the ordered family of C^∞-vector fields $(f_0,f_1,\dots,f_m)$ on X.

Definition II.1 (Network of robotic agents) *A* network of robotic agents *(or* robotic network*) $\mathcal{S}$ is a tuple $(I,\mathcal{A},E_{\text{cmm}})$ consisting of*

(i) $I=\{1,\dots,N\}$*; I is called the* set of unique identifiers (UIDs)*;*
(ii) $\mathcal{A}=\{A^{[i]}\}_{i\in I}=\{(X^{[i]},U^{[i]},X_0{}^{[i]},f^{[i]})\}_{i\in I}$ *is a set of control systems; this set is called the* set of physical agents*;*
(iii) E_{cmm} *is a map from $\prod_{i\in I}X^{[i]}$ to the subsets of $I\times I\setminus\operatorname{diag}(I\times I)$; this map is called the* communication edge map.

If $A^{[i]}=(X,U,X_0,f)$ for all $i\in I$, then the robotic network is called uniform. •

Let us comment on this definition and on how robotic agents communicate in a robotic network $(I,\mathcal{A},E_{\text{cmm}})$.

Remark II.2 By convention, we let the superscript $[i]$ denote the variables and spaces which correspond to the agent with unique identifier i; for instance, $x^{[i]}\in X^{[i]}$ and $x_0^{[i]}\in X_0^{[i]}$ denote the state and the initial state of agent $A^{[i]}$, respectively. We refer to $(x^{[1]},\dots,x^{[N]})\in\prod_{i\in I}X^{[i]}$ as a *state* of the network.

The map E_{cmm} models the topology of the communication service between the agents. In other words, at a network state $x=(x^{[1]},\dots,x^{[N]})$, two agents at locations $x^{[i]}$ and $x^{[j]}$ can communicate if the pair (i,j) is an edge in $E_{\text{cmm}}(x^{[1]},\dots,x^{[N]})$. Accordingly, we refer to the pair $(I,E_{\text{cmm}}(x^{[1]},\dots,x^{[N]}))$ as the *communication graph* at x. When and what agents communicate is discussed in Section II-B. Maps of the form $E\colon\prod_{i\in I}X^{[i]}\to 2^{I\times I\setminus\operatorname{diag}(I\times I)}$ are called *proximity edge maps*, and arise in wireless communication and computational geometry (see [1] for more details). Excluding edges of the form (i,i), $i\in I$, means that an individual agent does not communicate with itself. •

To make things concrete, let us present an interesting example of robotic network. Let $\mathbb{S}^1$ be the unit circle, and measure positions on $\mathbb{S}^1$ counterclockwise from the positive horizontal axis. For $x,y\in\mathbb{S}^1$, we let $\operatorname{dist}(x,y)=\min\{\operatorname{dist}_{\mathtt{c}}(x,y),\operatorname{dist}_{\mathtt{cc}}(x,y)\}$. Here, $\operatorname{dist}_{\mathtt{c}}(x,y)=(x-y)\pmod{2\pi}$ is the clockwise distance, that is, the path length from x to y traveling clockwise. Similarly, $\operatorname{dist}_{\mathtt{cc}}(x,y)=(y-x)\pmod{2\pi}$ is the counterclockwise distance. Here $x\pmod{2\pi}$ is the remainder of the division of x by 2π.

Example II.3 (Locally-connected first-order agents on the circle) For $r\in\mathbb{R}_+$, consider the uniform robotic network $\mathcal{S}_{\mathbb{S}^1,r\text{-disk}}=(I,\mathcal{A},E_{r\text{-disk}})$ composed of identical agents of the form $(\mathbb{S}^1,(0,\mathbf{e}))$. Here $\mathbf{e}$ is the vector field on $\mathbb{S}^1$ describing unit-speed counterclockwise rotation. We define the r-disk proximity edge map $E_{r\text{-disk}}$ on the circle by setting $(i,j)\in E_{r\text{-disk}}(\theta^{[1]},\dots,\theta^{[N]})$ if and only if

$$\operatorname{dist}(\theta^{[i]},\theta^{[j]})\le r\,,$$

where $\operatorname{dist}(x,y)$ is the geodesic distance between the two points x,y on the circle. •

B. Control and communication laws for robotic networks

Here we present a discrete-time communication, continuous-time motion model for the evolution of a robotic network. In our model, the robotic agents evolve in the physical domain in continuous-time and have the ability to exchange information (position and/or dynamic variables) that affect their motion at discrete-time instants.

Definition II.4 (Control and communication law) *Let $\mathcal{S}$ be a robotic network. A* (synchronous, dynamic, feedback) control and communication law $\mathcal{CC}$ *for $\mathcal{S}$ consists of the sets:*

(i) $\mathbb{T} = \{t_\ell\}_{\ell\in\mathbb{N}_0} \subset \overline{\mathbb{R}}_+$ *is an increasing sequence of time instants, called* communication schedule*;*
(ii) L *is a set containing the* `null` *element, called the* communication language*; elements of L are called* messages*;*
(iii) $W^{[i]}$, $i \in I$, *are sets of values of some* logic variables $w^{[i]}$, $i \in I$*;*
(iv) $W_0^{[i]} \subseteq W^{[i]}$, $i \in I$, *are subsets of* allowable initial values*;*

and of the maps:

(i) $\mathrm{msg}^{[i]} : \mathbb{T} \times X^{[i]} \times W^{[i]} \times I \to L$, $i \in I$, *are called* message-generation functions*;*
(ii) $\mathrm{stf}^{[i]} : \mathbb{T} \times W^{[i]} \times L^N \to W^{[i]}$, $i \in I$, *are called* state-transition functions*;*
(iii) $\mathrm{ctl}^{[i]} : \overline{\mathbb{R}}_+ \times X^{[i]} \times X^{[i]} \times W^{[i]} \times L^N \to U^{[i]}$, $i \in I$, *are called* control functions. •

We will sometimes refer to a control and communication law as a *motion coordination algorithm*. Control and communication laws might have various properties.

Definition II.5 (Properties of control and communication laws) *Let $\mathcal{S}$ be a robotic network and $\mathcal{CC}$ be a control and communication law for $\mathcal{S}$.*

(i) *If $\mathcal{S}$ is uniform and if $W^{[i]} = W$, $\mathrm{msg}^{[i]} = \mathrm{msg}$, $\mathrm{stf}^{[i]} = \mathrm{stf}$, $\mathrm{ctl}^{[i]} = \mathrm{ctl}$, for all $i \in I$, then $\mathcal{CC}$ is said to be* uniform *and is described by a tuple* $(\mathbb{T}, L, W, \{W_0^{[i]}\}_{i\in I}, \mathrm{msg}, \mathrm{stf}, \mathrm{ctl})$.
(ii) *If $W^{[i]} = W_0^{[i]} = \emptyset$ for all $i \in I$, then $\mathcal{CC}$ is said to be* static *and is described by a tuple* $(\mathbb{T}, L, \{\mathrm{msg}^{[i]}\}_{i\in I}, \{\mathrm{ctl}^{[i]}\}_{i\in I})$, *with* $\mathrm{msg}^{[i]} : \mathbb{T} \times X^{[i]} \times I \to L$, *and* $\mathrm{ctl}^{[i]} : \mathbb{T} \times X^{[i]} \times X^{[i]} \times L^N \to U^{[i]}$.
(iii) *$\mathcal{CC}$ is said to be* time-independent *if the message-generation, state-transition and control functions are of the form* $\mathrm{msg}^{[i]} : X^{[i]} \times W^{[i]} \times I \to L$, $\mathrm{stf}^{[i]} : W^{[i]} \times L^N \to W^{[i]}$, $\mathrm{ctl}^{[i]} : X^{[i]} \times X^{[i]} \times W^{[i]} \times L^N \to U^{[i]}$, $i \in I$, *respectively.* •

Roughly speaking this definition has the following meaning: for all $i \in I$, to the ith physical agent corresponds a logic process, labeled i, that performs the following actions. First, at each time instant $t_\ell \in \mathbb{T}$, the ith logic process sends to each of its neighbors in the communication graph a message (possibly the `null` message) computed by applying the message-generation function to the current values of $x^{[i]}$ and $w^{[i]}$. After a negligible period of time (therefore, still at time instant $t_\ell \in \mathbb{T}$), the ith logic process resets the value of its logic variables $w^{[i]}$ by applying the state-transition function to the current value of $w^{[i]}$, and to the messages received at time t_ℓ. Between communication instants, i.e., for $t \in [t_\ell, t_{\ell+1})$, the motion of the ith agent is determined by applying the control function to the current value of $x^{[i]}$, the value of $x^{[i]}$ at t_ℓ, and the current value of $w^{[i]}$. This idea is formalized as follows.

Definition II.6 (Evolution of a robotic network) *Let $\mathcal{S}$ be a robotic network and $\mathcal{CC}$ be a control and communication law for $\mathcal{S}$. The* evolution *of $(\mathcal{S}, \mathcal{CC})$ from initial conditions $x_0^{[i]} \in X_0{}^{[i]}$ and $w_0^{[i]} \in W_0^{[i]}$, $i \in I$, is the set of curves $x^{[i],\ell} : [t_\ell, t_{\ell+1}] \to X^{[i]}$, $i \in I$, $\ell \in \mathbb{N}_0$, and $w^{[i]} : \mathbb{T} \to W^{[i]}$, $i \in I$, satisfying*

$$\dot{x}^{[i],\ell}(t) = f\big(x^{[i],\ell}(t),\, \mathrm{ctl}^{[i]}(t, x^{[i],\ell}(t), x^{[i],\ell}(t_\ell), w^{[i]}(t_\ell), y^{[i]}(t_\ell))\big),$$

where, for $\ell \in \mathbb{N}_0$, and $i \in I$,

$$x^{[i],\ell}(t_\ell) = x^{[i],\ell-1}(t_\ell)\,, \quad w^{[i]}(t_\ell) = \mathrm{stf}^{[i]}(t_\ell, w^{[i]}(t_{\ell-1}), y^{[i]}(t_\ell))\,,$$

with the conventions that $x^{[i],-1}(t_0) = x_0^{[i]}$ and $w^{[i]}(t_{-1}) = w_0^{[i]}$, $i \in I$. Here, the function $y^{[i]} : \mathbb{T} \to L^N$ (describing the messages received by agent i) has components $y_j^{[i]}(t_\ell)$, for $j \in I$, given by

$$y_j^{[i]}(t_\ell) = \mathrm{msg}^{[j]}(t_\ell, x^{[j],\ell-1}(t_\ell), w^{[j]}(t_{\ell-1}), i)$$

if $(i, j) \in E_{\mathrm{cmm}}(x^{[1],\ell-1}(t_\ell), \dots, x^{[N],\ell-1}(t_\ell))$ and $y_j^{[i]}(t_\ell) =$ `null` otherwise. •

Remark II.7 (Idealized aspects of communication model) Let us discuss two limitations regarding the proposed communication model. We refer to $\mathcal{CC}$ as a *synchronous* control and communication law because the communications between all agents takes always place at the same time for all agents. We do not discuss here the important setting of asynchronous laws (see however Section IV).

The set L is used to exchange information between two robotic agents. The message `null` indicates no communication. We assume that the messages in the communication language L allow us to encode logical expressions such as `true` and `false`, integers, and real numbers. A realistic assumption on L would be to adopt a finite-precision representation for integers and real numbers in the messages. Instead, in what follows, we neglect any inaccuracies due to quantization (see however Section IV). •

Remark II.8 (Related notation) To distinguish between the `null` and the non-`null` messages received by an agent, it is convenient to define the *natural projection* $\pi_L : L^N \to 2^L$ that maps an array of messages y to the subset of L containing only the non-`null` messages in y.

In many uniform control and communication laws, the messages interchanged among the network agents are (quantized representations of) the agents' states and dynamic states. The corresponding communication language is $L = X \times W$ and message generation function $\mathrm{msg}_{\mathrm{std}} : \mathbb{T} \times X \times W \times I \to X \times W$, $\mathrm{msg}_{\mathrm{std}}(t, x, w, j) = (x, w)$, is referred to as the *standard message-generation function*.

By concatenating the curves $x^{[i],\ell}$ and $w^{[i],\ell}$, for $\ell \in \mathbb{N}_0$, we can define the evolution of the ith robotic agent $\overline{\mathbb{R}}_+ \ni t \mapsto (x^{[i]}(t), w^{[i]}(t)) \in X^{[i]} \times W^{[i]}$. Additionally we can define the curves

$$\overline{\mathbb{R}}_+ \ni t \mapsto x(t) = (x^{[1]}(t), \dots, x^{[N]}(t)) \in \prod_{i\in I} X^{[i]},$$

$$\overline{\mathbb{R}}_+ \ni t \mapsto w(t) = (w^{[1]}(t), \dots, w^{[N]}(t)) \in \prod_{i\in I} W^{[i]}. \quad \bullet$$

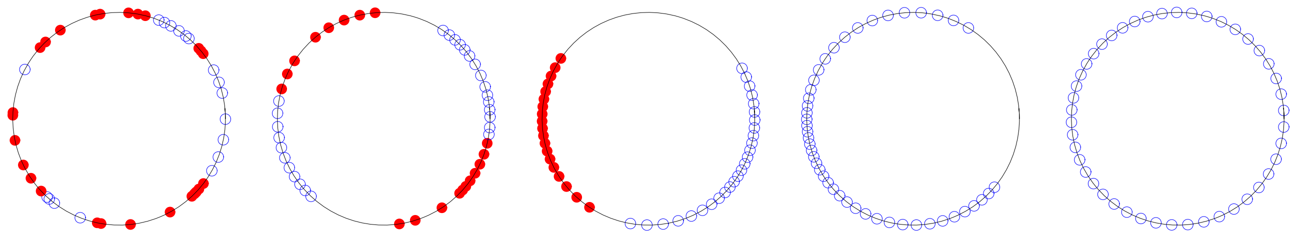

Fig. 1. The agree-and-pursue control and communication law in Section II-C with $N = 45$, $r = 2\pi/40$, and $k_{\text{prop}} = 1/4$. Disks and circles correspond to agents moving counterclockwise and clockwise, respectively. The initial positions and the initial directions of motion are randomly generated. The five pictures depict the network state at times $0, 12, 37, 100, 400$.

C. The agree-and-pursue control and communication law

From Example II.3, consider the uniform network $\mathcal{S}_{\mathbb{S}^1, r\text{-disk}}$ of locally-connected first-order agents in $\mathbb{S}^1$. We now define the agree-and-pursue law, denoted by $\mathcal{CC}_{\text{agr-pursuit}}$, as the uniform and time-independent law loosely described as follows:

> *[Informal description]* The dynamic variables are `drctn` taking values in $\{\texttt{c}, \texttt{cc}\}$ and `prior` taking values in I. At each communication round, each agent transmits its position and its dynamic variables and sets its dynamic variables to those of the incoming message with the largest value of `prior`. Between communication rounds, each agent moves in the counterclockwise or clockwise direction depending on whether its dynamic variable `drctn` is `cc` or `c`. For $k_{\text{prop}} \in]0, \frac{1}{2}[$, each agent moves k_{prop} times the distance to the immediately next neighbor in the chosen direction, or, if no neighbors are detected, k_{prop} times the communication range r.

Next, we define the law *formally*. Each agent has logic variables $w = (\texttt{drctn}, \texttt{prior})$, where $w_1 = \texttt{drctn} \in \{\texttt{cc}, \texttt{c}\}$, with arbitrary initial value, and $w_2 = \texttt{prior} \in I$, with initial value equal to the agent's identifier i. In other words, we define $W = \{\texttt{cc}, \texttt{c}\} \times I$, and we set $W_0^{[i]} = \{\texttt{cc}, \texttt{c}\} \times \{i\}$. Each agent $i \in I$ operates with the standard message-generation function, i.e., we set $L = \mathbb{S}^1 \times W$ and $\text{msg}^{[i]} = \text{msg}_{\text{std}}$, where $\text{msg}_{\text{std}}(\theta, w, j) = (\theta, w)$. The state-transition function is defined by

$$\text{stf}(w, y) = \text{argmax}\{z_2 \mid z \in (\pi_L(y))_2 \cup \{w\}\}.$$

For $k_{\text{prop}} \in \mathbb{R}_+$, the control function $\text{ctl}(\theta, \theta_{\text{smpld}}, w, y)$ is

$$k_{\text{prop}} \min\left(\{r\} \cup \{\text{dist}_{\texttt{cc}}(\theta_{\text{smpld}}, \theta_{\text{rcvd}}) \mid \theta_{\text{rcvd}} \in (\pi_L(y))_1\}\right)$$

if $\texttt{drctn} = \texttt{cc}$, and

$$-k_{\text{prop}} \min\left(\{r\} \cup \{\text{dist}_{\texttt{c}}(\theta_{\text{smpld}}, \theta_{\text{rcvd}}) \mid \theta_{\text{rcvd}} \in (\pi_L(y))_1\}\right)$$

if $\texttt{drctn} = \texttt{c}$.

Finally, we sketch the control and communication in equivalent pseudocode language. This is possible for this example, and necessary for more complicated ones. For example, the state-transition function is written as:

```
function stf((drctn, prior), y)
for each non-null message
        (θ_rcvd, (drctn_rcvd, prior_rcvd)) in y:
  if (prior_rcvd > prior), then
    drctn := drctn_rcvd
    prior := prior_rcvd
  endif
endfor
return (drctn,prior)
```

Similarly, the control function ctl is written as:

```
function ctl(θ, θ_smpld, (drctn, prior), y)
d_tmp := r
for each non-null message
        (θ_rcvd, (drctn_rcvd, prior_rcvd)) in y:
  if (drctn = cc) AND (dist_cc(θ_smpld, θ_rcvd) < d_tmp),
    then d_tmp := dist_cc(θ_smpld, θ_rcvd)
  elseif (drctn = c) AND (dist_c(θ_smpld, θ_rcvd) < d_tmp),
    then d_tmp := dist_c(θ_smpld, θ_rcvd)
  endif
endfor
if (drctn = cc), then return k_prop d_tmp,
  else return −k_prop d_tmp endif
```

An implementation of this control and communication law is shown in Fig. 1. Note that, along the evolution, all agents agree upon a common direction of motion and, after suitable time, they reach a uniform distribution. Finally, we remark that this law is related to the leader election algorithm discussed in [4]. •

III. Coordination tasks and complexity measures

In this section we introduce concepts and tools useful to analyze a communication and control law. We address the following issues: What is a coordination task for a robotic network? When does a control and communication law achieve a task? And with what time and communication complexity?

A. Coordination tasks

Our first analysis step is to characterize the correctness properties of a communication and control law. We do so

by defining the notion of task and of task achievement by a robotic network.

Definition III.1 (Coordination task) *Let $\mathcal{S}$ be a robotic network and let $\mathcal{W}$ be a set.*

(i) *A* coordination task *for $\mathcal{S}$ is a map $\mathcal{T}\colon \prod_{i\in I} X^{[i]} \times \mathcal{W}^N \to \texttt{BooleSet}$.*
(ii) *If $\mathcal{W} = \emptyset$, then the coordination task is said to be* static *and is described by a map $\mathcal{T}\colon \prod_{i\in I} X^{[i]} \to \texttt{BooleSet}$.*

Additionally, let $\mathcal{CC}$ a control and communication law for $\mathcal{S}$.

(i) *The law $\mathcal{CC}$ is* compatible *with the task $\mathcal{T}\colon \prod_{i\in I} X^{[i]} \times \mathcal{W}^N \to \texttt{BooleSet}$ if its logic variables take values in $\mathcal{W}$, that is, if $W^{[i]} = \mathcal{W}$, for all $i \in I$.*
(ii) *The law $\mathcal{CC}$* achieves *the task $\mathcal{T}$ if it is compatible with it and if, for all initial conditions $x_0^{[i]} \in X_0^{[i]}$ and $w_0^{[i]} \in W_0^{[i]}$, $i \in I$, the corresponding network evolution $t \mapsto (x(t), w(t))$ has the property that there exists $T \in \mathbb{R}_+$ such that $\mathcal{T}(x(t), w(t)) = \texttt{true}$ for all $t \geq T$.* •

Loosely speaking, achieving a task might mean obtaining a specified pattern in the position of the agents or of their dynamic variables.

Example III.2 (Agreement and equidistance tasks) From Example II.3, consider the uniform network $\mathcal{S}_{\mathbb{S}^1,r\text{-disk}}$ of locally-connected first-order agents in $\mathbb{S}^1$. From Example II-C, recall the agree-and-pursue control and communication law $\mathcal{CC}_{\text{agr-pursuit}}$ with dynamic variables taking values in $W = \{\texttt{cc}, \texttt{c}\} \times I$. There are two tasks of interest. First, we define the *agreement task* $\mathcal{T}_{\texttt{drctn}}\colon (\mathbb{S}^1)^N \times W^N \to \texttt{BooleSet}$ by

$$\mathcal{T}_{\texttt{drctn}}(\theta, w) = \begin{cases} \texttt{true}, & \text{if } \texttt{drctn}^{[1]} = \dots = \texttt{drctn}^{[N]}, \\ \texttt{false}, & \text{otherwise}, \end{cases}$$

where $\theta = (\theta^{[1]}, \dots, \theta^{[N]})$, $w = (w^{[1]}, \dots, w^{[N]})$, and $w^{[i]} = (\texttt{drctn}^{[i]}, \texttt{prior}^{[i]})$, for $i \in I$. Furthermore, for $\varepsilon > 0$, we define the static *ε-equidistance task* $\mathcal{T}_{\text{eqdstnc}}\colon (\mathbb{S}^1)^N \to \texttt{BooleSet}$ by $\mathcal{T}_{\varepsilon\text{-eqdstnc}}(\theta) = \texttt{true}$ if and only if

$$\big| \min_{j\neq i} \operatorname{dist}_{\texttt{c}}(\theta^{[i]}, \theta^{[j]}) - \min_{j\neq i} \operatorname{dist}_{\texttt{cc}}(\theta^{[i]}, \theta^{[j]}) \big| < \varepsilon, \text{ for all } i \in I.$$

In other words, $\mathcal{T}_{\varepsilon\text{-eqdstnc}}$ is true when, for every agent, the clockwise distance to the closest clockwise neighbor and the counterclockwise distance to the closest counterclockwise neighbor are approximately equal. •

B. Complexity notions for control and communication laws and for coordination tasks

We are finally ready to define the key notions of time and communication complexity. These notions describe the cost that a certain control and communication law incurs while completing a certain coordination task. We also define the complexity of a task to be the infimum of the costs incurred by all laws that achieve that task.

First we define the time complexity of an achievable task as the minimum number of communication rounds needed by the agents to achieve the task $\mathcal{T}$.

Definition III.3 (Time complexity) *Let $\mathcal{S}$ be a robotic network and let $\mathcal{T}$ be a coordination task for $\mathcal{S}$. Let $\mathcal{CC}$ be a control and communication law for $\mathcal{S}$ compatible with $\mathcal{T}$.*

(i) *The* time complexity to achieve $\mathcal{T}$ with $\mathcal{CC}$ from $(x_0, w_0) \in \prod_{i\in I} X_0^{[i]} \times \prod_{i\in I} W_0^{[i]}$ *is*
$$\operatorname{TC}(\mathcal{T}, \mathcal{CC}, x_0, w_0) = \inf \{\ell \mid \mathcal{T}(x(t_k), w(t_k)) = \texttt{true}, \text{ for all } k \geq \ell\},$$
where $t \mapsto (x(t), w(t))$ is the evolution of $(\mathcal{S}, \mathcal{CC})$ from the initial condition (x_0, w_0).
(ii) *The* time complexity to achieve $\mathcal{T}$ with $\mathcal{CC}$ *is*
$$\operatorname{TC}(\mathcal{T}, \mathcal{CC}) = \sup \Big\{ \operatorname{TC}(\mathcal{T}, \mathcal{CC}, x_0, w_0) \mid (x_0, w_0) \in \prod_{i\in I} X_0^{[i]} \times \prod_{i\in I} W_0^{[i]} \Big\}.$$
(iii) *The* time complexity of $\mathcal{T}$ *is*
$$\operatorname{TC}(\mathcal{T}) = \inf\{\operatorname{TC}(\mathcal{T}, \mathcal{CC}) \mid \mathcal{CC} \text{ compatible with } \mathcal{T}\}. \quad \bullet$$

Next, we define the notion of mean and total communication complexities for a task. As usual, we assume that the network $\mathcal{S}$ has a communication edge map E_{cmm} and that the control and communication law $\mathcal{CC}$ has language L and message-generation functions $\operatorname{msg}^{[i]}$, $i \in I$. With these data we can discuss the communication cost of realizing one communication round. At time $t \in \mathbb{T}$ from state $(x, w) \in \prod_{i\in I} X^{[i]} \times \prod_{i\in I} W^{[i]}$, an element of L needs to be transmitted for each edge of the directed graph $(I, E_{\text{cmm}\setminus\emptyset}(t, x, w))$ defined by $(i, j) \in E_{\text{cmm}\setminus\emptyset}(t, x, w)$ if and only if

$$(i, j) \in E_{\text{cmm}}(x) \text{ and } \operatorname{msg}^{[i]}(t, x^{[i]}, w^{[i]}, j) \neq \texttt{null}.$$

Next, we need a model for the cost of sending a message for each directed edge in $E_{\text{cmm}\setminus\emptyset}$.

Definition III.4 (One-round cost) *For $I = \{1, \dots, N\}$, a function $\mathrm{C}_{\text{rnd}}\colon 2^{I\times I} \to \overline{\mathbb{R}}_+$ is a* one-round cost function *if $\mathrm{C}_{\text{rnd}}(\emptyset) = 0$, and $S_1 \subset S_2 \subset I \times I$ implies $\mathrm{C}_{\text{rnd}}(S_1) \leq \mathrm{C}_{\text{rnd}}(S_2)$. A one-round cost function C_{rnd} is* additive *if, for all $S_1, S_2 \subset I \times I$, $S_1 \cap S_2 = \emptyset$ implies $\mathrm{C}_{\text{rnd}}(S_1 \cup S_2) = \mathrm{C}_{\text{rnd}}(S_1) + \mathrm{C}_{\text{rnd}}(S_2)$.* •

This definition is motivated by the assumptions that (i) the cost of exchanging any message is bounded, and that (ii) this cost is zero only for the `null` message. More specific detail about the communication cost depends necessarily on the type of communication service (e.g. unidirectional versus omnidirectional) available between the agents (see [1] for an extended discussion). Here we only emphasize that, for a given control and communication law $\mathcal{CC}$ with language L, the one-round cost depends on L; we therefore write it as $\mathrm{C}^L_{\text{rnd}}\colon 2^{I\times I} \to \overline{\mathbb{R}}_+$.

Definition III.5 (Communication complexity) *Let $\mathcal{S}$ be a robotic network and let $\mathcal{T}$ be a coordination task for $\mathcal{S}$. Let $\mathcal{CC}$ be a control and communication law for $\mathcal{S}$ compatible with $\mathcal{T}$, and let $\mathrm{C}^L_{\mathrm{rnd}} \colon 2^{I\times I} \to \overline{\mathbb{R}}_+$ be a one-round cost function.*

(i) *The* mean communication complexity *and the* total communication complexity *to achieve $\mathcal{T}$ with $\mathcal{CC}$ from $(x_0, w_0) \in \prod_{i\in I} X_0^{[i]} \times \prod_{i\in I} W_0^{[i]}$ are, respectively,*

$$\mathrm{MCC}(\mathcal{T}, \mathcal{CC}, x_0, w_0) = \frac{1}{\lambda}\sum_{\ell=0}^{\lambda-1} \mathrm{C}^L_{\mathrm{rnd}} \circ E_{\mathrm{cmm}\setminus\emptyset}(t_\ell, x(t_\ell), w(t_\ell)),$$

$$\mathrm{TCC}(\mathcal{T}, \mathcal{CC}, x_0, w_0) = \sum_{\ell=0}^{\lambda-1} \mathrm{C}^L_{\mathrm{rnd}} \circ E_{\mathrm{cmm}\setminus\emptyset}(t_\ell, x(t_\ell), w(t_\ell)),$$

where $\lambda = \mathrm{TC}(\mathcal{CC}, \mathcal{T}, x_0, w_0)$ and $t \mapsto (x(t), w(t))$ is the evolution of $(\mathcal{S}, \mathcal{CC})$ from the initial condition (x_0, w_0). (Here MCC *is defined only for (x_0, w_0) with the property that $\mathcal{T}(x_0, w_0) =$* `false`*.)*

(ii) *The* mean communication complexity *and the* total communication complexity *to achieve $\mathcal{T}$ with $\mathcal{CC}$ are the supremum of $\{\mathrm{MCC}(\mathcal{T}, \mathcal{CC}, x_0, w_0) \mid (x_0, w_0) \in \prod_{i\in I} X_0^{[i]} \times \prod_{i\in I} W_0^{[i]}\}$ and $\{\mathrm{TCC}(\mathcal{T}, \mathcal{CC}, x_0, w_0) \mid (x_0, w_0) \in \prod_{i\in I} X_0^{[i]} \times \prod_{i\in I} W_0^{[i]}\}$, respectively.*

(iii) *The* mean communication complexity of $\mathcal{T}$ *and the* total communication complexity of $\mathcal{T}$ *are, respectively,*

$$\mathrm{MCC}(\mathcal{T}) = \inf\{\mathrm{MCC}(\mathcal{T}, \mathcal{CC}) \mid \mathcal{CC} \textit{ compatible with } \mathcal{T}\},$$
$$\mathrm{TCC}(\mathcal{T}) = \inf\{\mathrm{TCC}(\mathcal{T}, \mathcal{CC}) \mid \mathcal{CC} \textit{ compatible with } \mathcal{T}\}.$$

•

It is clear that, for $(x_0, w_0) \in \prod_{i\in I} X_0^{[i]} \times \prod_{i\in I} W_0^{[i]}$,

$$\mathrm{TCC}(\mathcal{T}, \mathcal{CC}, x_0, w_0) = \mathrm{MCC}(\mathcal{T}, \mathcal{CC}, x_0, w_0) \cdot \mathrm{TC}(\mathcal{T}, \mathcal{CC}, x_0, w_0).$$

This implies that $\mathrm{TCC}(\mathcal{T}, \mathcal{CC}) \le \mathrm{MCC}(\mathcal{T}, \mathcal{CC}) \cdot \mathrm{TC}(\mathcal{T}, \mathcal{CC})$.

C. Invariance under rescheduling of control and communication laws

In this section, we discuss the invariance properties of the notions of time and communication complexity under the *rescheduling* of a control and communication law. The idea behind rescheduling is to "spread" the execution of the law over time without affecting the trajectories described by the robotic agents of the network.

Let $\mathcal{S} = (I, \mathcal{A}, E_{\mathrm{cmm}})$ be a robotic network with driftless physical agents, that is, a robotic network where each physical agent is a driftless control system. Let $\mathcal{CC} = (\mathbb{N}_0, L, \{\mathrm{ctl}^{[i]}\}_{i\in I}, \{\mathrm{msg}^{[i]}\}_{i\in I})$ be a static control and communication law. It is out intention to define a new control and communication law by modifying $\mathcal{CC}$; to do so we introduce some notation. Let $s \in \mathbb{N}$, with $s \le N$, and let $\mathcal{P}_I = \{I_0, \dots, I_{s-1}\}$ be an s-*partition* of I, that is, $I_0, \dots, I_{s-1}$ are disjoint and nonempty subsets of I and $I = \cup_{k=0}^{s-1} I_k$.

For $i \in I$, define the message-generation functions $\mathrm{msg}^{[i]}_{(s,\mathcal{P}_I)} \colon \mathbb{N}_0 \times X^{[i]} \times I \to L$ by

$$\mathrm{msg}^{[i]}_{(s,\mathcal{P}_I)}(t_\ell, x, j) = \mathrm{msg}^{[i]}(t_{\lfloor \ell/s \rfloor}, x, j), \tag{1}$$

if $i \in I_k$ and $k = \ell (\mathrm{mod}\, s)$, and $\mathrm{msg}^{[i]}_{(s,\mathcal{P}_I)}(t_\ell, x, j) =$ `null` otherwise. According to this new message-generation function, only the agents with unique identifier in I_k will send messages at time t_ℓ, with $\ell \in \{k + as \mid a \in \mathbb{N}_0\}$. Equivalently, this can be stated as follows. Define the increasing function $F \colon \mathbb{N}_0 \to \mathbb{N}_0$ by $F(\ell) = s(\ell+1) - 1$. According to the message-generation functions specified by (1), the messages originally sent at the time instant t_ℓ are now rescheduled to be sent at the time instants $t_{F(\ell)-s+1}, \dots, t_{F(\ell)}$. Fig. 2 illustrates this idea.

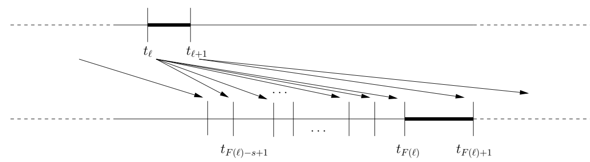

Fig. 2. Under the rescheduling, the messages that are sent at the time instant t_ℓ under the control and communication law $\mathcal{CC}$ are rescheduled to be sent over the time instants $t_{F(\ell)-s+1}, \dots, t_{F(\ell)}$ under the control and communication law $\mathcal{CC}_{(s,\mathcal{P}_I)}$. Accordingly, the evolution of the robotic network under the original law during the time interval $[t_\ell, t_{\ell+1}]$ is now executed when all the corresponding messages have been transmitted, i.e., along the time interval $[t_{F(\ell)}, t_{F(\ell)+1}]$.

For $i \in I$, define the control functions $\mathrm{ctl}^{[i]} \colon \overline{\mathbb{R}}_+ \times X^{[i]} \times X^{[i]} \times L^N \to U^{[i]}$ by

$$\mathrm{ctl}^{[i]}_{(s,\mathcal{P}_I)}(t, x, x_{\mathrm{smpld}}, y) = \frac{t_{F^{-1}(\ell)+1} - t_{F^{-1}(\ell)}}{t_{\ell+1} - t_\ell}\, \mathrm{ctl}^{[i]}\left(h_\ell(t), x, x_{\mathrm{smpld}}, y\right), \tag{2}$$

if $t \in [t_\ell, t_{\ell+1}]$ and $\ell = -1 (\mathrm{mod}\, s)$ and $\mathrm{ctl}^{[i]}_{(s,\mathcal{P}_I)}(t, x, x_{\mathrm{smpld}}, y) = 0$ otherwise. Here $F^{-1} \colon \mathbb{N}_0 \to \mathbb{N}_0$ is the inverse of F, defined by $F^{-1}(\ell) = \frac{\ell+1}{s} - 1$, and for $\ell = -1(\mathrm{mod}\, s)$, the function $h_\ell \colon [t_\ell, t_{\ell+1}] \to [t_{F^{-1}(\ell)}, t_{F^{-1}(\ell)+1}]$ is the time re-parameterization function defined by

$$h_\ell(t) = \frac{(t_{F^{-1}(\ell)+1} - t_{F^{-1}(\ell)})t + t_{\ell+1} t_{F^{-1}(\ell)} - t_\ell\, t_{F^{-1}(\ell)+1}}{t_{\ell+1} - t_\ell},$$

for $[t_\ell, t_{\ell+1}]$. Roughly speaking, the control law $\mathrm{ctl}^{[i]}_{(s,\mathcal{P}_I)}$ makes the agent i wait for the time intervals $[t_\ell, t_{\ell+1}]$, with $\ell \in \{as - 1 \mid a \in \mathbb{N}\}$, to execute any motion. Accordingly, the evolution of the robotic network under the original law $\mathcal{CC}$ during the time interval $[t_\ell, t_{\ell+1}]$ now takes place when all the corresponding messages have been transmitted, i.e., along the time interval $[t_{F(\ell)}, t_{F(\ell)+1}]$.

This construction is gathered in the following definition.

Definition III.6 (Rescheduling of control and communication laws) *Let $\mathcal{S} = (I, \mathcal{A}, E_{\mathrm{cmm}})$ be a robotic network with driftless physical agents, and let $\mathcal{CC} = (\mathbb{N}_0, L, \{\mathrm{ctl}^{[i]}\}_{i\in I}, \{\mathrm{msg}^{[i]}\}_{i\in I})$ be a static control and communication law. Let $s \in \mathbb{N}$, with $s \leq N$, and let $\mathcal{P}_I$ be an s-partition of I. The control and communication law $\mathcal{CC}_{(s,\mathcal{P}_I)} = (\mathbb{N}_0, L, \{\mathrm{ctl}^{[i]}_{(s,\mathcal{P}_I)}\}_{i\in I}, \{\mathrm{msg}^{[i]}_{(s,\mathcal{P}_I)}\}_{i\in I})$ defined by equations* (1) *and* (2) *is called a $(s, \mathcal{P}_I)$*-rescheduling of $\mathcal{CC}$. •

Next, we examine the relation between the evolutions and the time and communication complexities of a control and communication law, and of those of its reschedulings. The following result shows that the total communication cost of a $\mathcal{CC}$ remains invariant under rescheduling. The proof can be found in [1].

Theorem III.7 *With the same assumptions as in Definition III.6, let $t \mapsto x(t)$ and $t \mapsto \tilde{x}(t)$ denote the network evolutions starting from $x_0 \in \prod_{i\in I} X_0^{[i]}$ under $\mathcal{CC}$ and $\mathcal{CC}_{(s,\mathcal{P}_I)}$, respectively, and let $\mathcal{T}$: $\prod_{i\in I} X^{[i]} \to \mathtt{BooleSet}$ be a coordination task for $\mathcal{S}$. For all $k \in \mathbb{N}_0$,*

$$\tilde{x}^{[i]}(t) = \begin{cases} \tilde{x}^{[i]}(t_{F(k-1)+1}), & \text{for } t \in \bigcup_{\ell=F(k-1)+1}^{F(k)-1} [t_\ell, t_{\ell+1}]\,, \\ x^{[i]}(h_{F(k)}(t)), & \text{for } t \in [t_{F(k)}, t_{F(k)+1}]\,. \end{cases} \tag{3}$$

For all $x_0 \in \prod_{i\in I} X_0^{[i]}$,

$$\mathrm{TC}(\mathcal{CC}_{(s,\mathcal{P}_I)}, \mathcal{T}, x_0) = s \cdot \mathrm{TC}(\mathcal{CC}, \mathcal{T}, x_0)\,.$$

If $\mathrm{C}_{\mathrm{rnd}}$ is additive, then, for all $x_0 \in \prod_{i\in I} X_0^{[i]}$

$$\mathrm{MCC}(\mathcal{CC}_{(s,\mathcal{P}_I)}, \mathcal{T}, x_0) = \frac{1}{s} \cdot \mathrm{MCC}(\mathcal{CC}, \mathcal{T}, x_0)\,,$$

and, therefore, the total communication cost of $\mathcal{CC}$ is invariant under rescheduling. •

D. Agreement on direction of motion and equidistance

From Examples II.3, II-C and III.2, recall the definition of uniform network $\mathcal{S}_{\mathbb{S}^1, r\text{-disk}}$ of locally-connected first-order agents in $\mathbb{S}^1$, the agree-and-pursue control and communication law $\mathcal{CC}_{\text{agr-pursuit}}$, and the two coordination tasks $\mathcal{T}_{\mathtt{drctn}}$ and $\mathcal{T}_{\varepsilon\text{-eqdstnc}}$. The following result characterizes the complexity to achieve these coordination tasks with $\mathcal{CC}_{\text{agr-pursuit}}$. The proof can be found in [1].

Theorem III.8 *For $k_{prop} \in]0, \frac{1}{2}[$, $r \in]0, 2\pi]$, $\alpha = Nr - 2\pi$ and $\varepsilon \in]0, 1[$, the network $\mathcal{S}_{\mathbb{S}^1, r\text{-disk}}$, the law $\mathcal{CC}_{\text{agr-pursuit}}$, and the tasks $\mathcal{T}_{\mathtt{drctn}}$ and $\mathcal{T}_{\varepsilon\text{-eqdstnc}}$ together satisfy:*

(i) *the upper bound $\mathrm{TC}(\mathcal{T}_{\mathtt{drctn}}, \mathcal{CC}_{\text{agr-pursuit}}) \in O(Nr^{-1})$ and the lower bound*

$$\mathrm{TC}(\mathcal{T}_{\mathtt{drctn}}, \mathcal{CC}_{\text{agr-pursuit}}) \in \begin{cases} \Omega(r^{-1}) & \text{if } \alpha \geq 0, \\ \Omega(N) & \text{if } \alpha \leq 0; \end{cases}$$

(ii) *if $\alpha > 0$, the upper bound $\mathrm{TC}(\mathcal{T}_{\varepsilon\text{-eqdstnc}}, \mathcal{CC}_{\text{agr-pursuit}}) \in O(N^2 \log(N\varepsilon^{-1}) + N \log \alpha^{-1})$ and the lower bound $\mathrm{TC}(\mathcal{T}_{\varepsilon\text{-eqdstnc}}, \mathcal{CC}_{\text{agr-pursuit}}) \in \Omega(N^2 \log(\varepsilon^{-1}))$. If $\alpha \leq 0$, then $\mathcal{CC}_{\text{agr-pursuit}}$ does not achieve $\mathcal{T}_{\varepsilon\text{-eqdstnc}}$ in general.*

IV. Conclusions

We have introduced a formal model for the design and analysis of coordination algorithms executed by networks composed of robotic agents. Under this framework, motion coordination algorithms are formalized as feedback control and communication laws. Drawing analogies with the classical theory on distributed algorithms, we have defined two measures of complexity for this formal notion of execution: the time and the mean communication complexity of achieving a specific task. We have defined the notion of re-scheduling of a control and communication law and analyzed the invariance of the proposed complexity measures under this operation. These concepts and results have been illustrated in a network of locally connected agents on the circle executing the "agree-and-pursue" motion coordination algorithm.

Numerous avenues for future research appear open. An incomplete list include the following: (i) modeling of asynchronous networks (see however [16], [17], [10], [18]); (ii) models and analysis of failures in the agents (arrivals/departures) and in the communication links (see however [19], [20], [21], [22]); (iii) probabilistic versions of the complexity measures (see however [12]); (iv) quantization and delays in the communication channels (see however [23] and the literature on quantized control); and (v) parallel, sequential and hierarchical composition of control and communication laws. On the algorithmic side, the companion paper [2] provides time-complexity estimates for various coordination algorithms that achieve rendezvous and deployment, and discusses other open questions.

Acknowledgments

This material is based upon work supported in part by ONR YIP Award N00014-03-1-0512, NSF SENSORS Award IIS-0330008, and NSF CAREER Award CCR-0133869. Sonia Martínez's work was supported in part by a Fulbright Postdoctoral Fellowship from the Spanish Ministry of Education and Science.

References

[1] S. Martínez, F. Bullo, J. Cortés, and E. Frazzoli, "Synchronous robotic networks and complexity of control and communication laws," *Preprint. Available electronically at* `http://xxx.arxiv.org/math.OC/0501499`, Jan. 2005.

[2] ——, "On synchronous robotic networks – Part II: Time complexity of rendezvous and deployment algorithms," in *IEEE Conf. on Decision and Control*, Seville, Spain, Feb. 2005, submitted.

[3] Committee on Networked Systems of Embedded Computers, *Embedded, Everywhere: A Research Agenda for Networked Systems of Embedded Computers*. National Academy Press, 2001.

[4] N. A. Lynch, *Distributed Algorithms*. San Mateo, CA: Morgan Kaufmann Publishers, 1997.

[5] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Belmont, MA: Athena Scientific, 1997.

[6] I. Suzuki and M. Yamashita, "Distributed anonymous mobile robots: Formation of geometric patterns," *SIAM Journal on Computing*, vol. 28, no. 4, pp. 1347–1363, 1999.

[7] H. Ando, Y. Oasa, I. Suzuki, and M. Yamashita, "Distributed memoryless point convergence algorithm for mobile robots with limited visibility," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 5, pp. 818–828, 1999.

[8] K. Sugihara and I. Suzuki, “Distributed algorithms for formation of geometric patterns with many mobile robots,” *Journal of Robotic Systems*, vol. 13, no. 3, pp. 127–39, 1996.

[9] P. Flocchini, G. Prencipe, N. Santoro, and P. Widmayer, “Hard tasks for weak robots: The role of common knowledge in pattern formation by autonomous mobile robots,” in *ISAAC 1999, 10th International Symposium on Algorithm and Computation (Chennai, India)*, ser. Lecture Notes in Computer Science, A. Aggarwal and C. P. Rangan, Eds. New York: Springer Verlag, 1999, vol. 1741, pp. 93–102.

[10] ——, “Gathering of asynchronous oblivious robots with limited visibility,” *Theoretical Computer Science*, 2005, to appear.

[11] N. Santoro, “Distributed computations by autonomous mobile robots,” in *SOFSEM 2001: Conference on Current Trends in Theory and Practice of Informatics (Piestany, Slovak Republic)*, ser. Lecture Notes in Computer Science, L. Pacholski and P. Ruzicka, Eds. New York: Springer Verlag, 2001, vol. 2234, pp. 110–115.

[12] E. Klavins, “Communication complexity of multi-robot systems,” in *Algorithmic Foundations of Robotics V*, ser. STAR, Springer Tracts in Advanced Robotics, J.-D. Boissonnat, J. W. Burdick, K. Goldberg, and S. Hutchinson, Eds., vol. 7. Berlin Heidelberg: Springer Verlag, 2003.

[13] ——, “A computation and control language for multi-vehicle systems,” in *IEEE Conf. on Decision and Control*, Maui, Hawaii, Dec. 2003, pp. 4133–4139.

[14] N. A. Lynch, R. Segala, and F. Vaandrager, “Hybrid I/O automata,” *Information and Computation*, vol. 185, no. 1, pp. 105–157, 2003.

[15] M. Savchenko and E. Frazzoli, “On the time complexity of the multiple-vehicle coordination problem,” in *American Control Conference*, Portland, OR, Sept. 2004, to appear.

[16] J. Lin, A. S. Morse, and B. D. O. Anderson, “The multi-agent rendezvous problem - the asynchronous case,” in *IEEE Conf. on Decision and Control*, Paradise Island, Bahamas, Dec. 2004, pp. 1926–1931.

[17] J. Cortés, S. Martínez, T. Karatas, and F. Bullo, “Coverage control for mobile sensing networks,” *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.

[18] X. Defago and A. Konagaya, “Circle formation for oblivious anonymous mobile robots with no common sense of orientation,” in *ACM International Workshop on Principles of Mobile Computing (POMC 02)*, Toulouse, France, Oct. 2002, pp. 97–104.

[19] J. Cortés, S. Martínez, and F. Bullo, “Robust rendezvous for mobile autonomous agents via proximity graphs in d dimensions,” *IEEE Transactions on Automatic Control*, June 2004, submitted. Electronic version available at `http://www.me.ucsb.edu/bullo`.

[20] R. Olfati-Saber and R. M. Murray, “Consensus problems in networks of agents with switching topology and time-delays,” *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[21] L. Moreau, “Time-dependent unidirectional communication in multi-agent systems,” *IEEE Transactions on Automatic Control*, 2004, to appear. Electronically available at `http://xxx.arxiv.org/math.OC/0306426`.

[22] W. Ren and R. W. Beard, “Consensus seeking in multi-agent systems using dynamically changing interaction topologies,” *IEEE Transactions on Automatic Control*, Dec. 2003, to appear.

[23] F. Fagnani, K. H. Johansson, A. Speranzon, and S. Zampieri, “On multi-vehicle rendezvous under quantized communication,” in *Mathematical Theory of Networks and Systems*, Leuven, Belgium, July 2004, electronic proceedings.